# Defending Against Out-of-Band Management BMC Attacks

Lee Fisher
April 2019
LinuxFest NorthWest

"Imagine trying to secure a computer with a small but powerful parasitic server on its motherboard; a bloodsucking leech that can't be turned off and has no documentation; you can't login, patch, or fix problems on it; server-based defensive, audit, or anti-malware software can’t be used for protection; its design is secret, implementation old, and it can fully control the computer's hardware and software; and it shares passwords with a bunch of other important servers, stores them in clear text for attackers to access. Not to mention it was designed for full control, remote management and monitoring, and it’s pretty damn good at it."

--Dan Farmer, 2013

# Agenda

- BMC/LOM concepts
- MC/SP (Intel ME/AMT, AMD PSP, Apple T2, ...)
- IPMI
- Intel SMM
- WS-MAN
- SMASH and DASH
- OpenBMC
- Redfish
- “IPMI++” (HP iLO, Dell iDRAC, ….)

# Credits

- All security guidance in this talk comes from existing BMC security research from:
- Dan Farmer, HD Moore, Matias Soler, Nicolas Waisman, Fabien Périgaud, Alexandre Gazet, Joffrey Czarny, Adrien Guinet, Jesse Michael, Mickey Shkatov, Oleksandr Bazhaniuk,

  ...and others that I am forgetting (sorry)

# CPU (and SMM)

- In early systems, the CPU was in charge of everything. Via the CPU, the firmware and OS code talked to the registers, RAM, and I/O busses. The CPU was in charge of system security.
  - This is the traditional model that attackers use, OS/app-level malware.
- On modern Intel (and AMD) systems, in addition to normal CPU mode, the CPU has a new mode, SMM (Systems Management Mode). SMM code can see all the normal CPU can see, plus SMBIOS and SMRAM, which normal CPU modes cannot see. See later slide on SMM and IPMI.
  - Attackers are also using SMM.
  - The rough equivalent of Intel SMM on ARM is TrustZone,
    both are called “Management Mode” from UEFI firmware perspective

# Baseboard Management Controller (BMC)

- On modern x64 server systems, the CPU is no longer in charge of system security, a new BMC chip is
- BMC is a separate chip beyond the CPU, which can see all the CPU's resources (but the CPU cannot see the BMC)
- BMCs are mostly in server-based systems, but also in some ["business class"] desktop/mobile systems (eg, see DASH and Intel AMT/ME)
- An embedded device with an independent microprocessor used to perform systems monitoring and management-related tasks on a computer system, in-band or out-of-band

# Lights-Out Management (LOM)

- Convenience features added to help sysadmins remotely administer systems, mostly servers but also enterprise desktop/laptops. LOM are the core features of a BMC.
- Attack surface which can be used by to remotely attack systems
- Standardized in (IPMI, SMASH, DASH, Redfish, …)
- Vendors include vendor-centric services/protocols and tools
- Can be used in-bound or out-of-bound

# Out-of-Bounds Management

- Normal computing is done with using main CPU, as result of end-user booting an OS and running an app (“OS-present app”). We think the OS is talking to the CPU to run the app
  - Or perhaps instead of an OS-present app, using a “pre-OS app”, like a UEFI Shell command line tool. Firmware is talking to the CPU to run the app
- Out-of-Bounds Management is when using BMC to control system outside the scope of the main CPU. OOB mgmt works when computer is ‘powered off’ (CPU is not running)
- Most operate remotely via Ethernet, but some use WiFi

# Intel ME/AMT

- The Intel ME chip and it’s AMT software is used by Intel for multiple things, including being a LOM chip
- Like SMM and BMC, Intel ME is mostly invisible to CPU
- Unlike most BMC usage, Intel ME is on most desktops
- AMD systems have the AMD PSP (Platform Security Processor) chip
- I believe new Apple systems have the Intel ME chip as well as the new Apple T2 processor

# Management Controllers (Security Processors)

- In addition to CPU, and BMC, modern systems (all x64 systems, not just servers) also now have a Management Controller or Security Processor, which is not strictly BMC/LOM-focused but some have some BMC features (secure the boot process from attacks, invisible management tasks in background)
  - Examples: Intel ME/AMT, AMD PSP, Apple T2
- This is a separate chip from CPU and BMC, with an embedded OS and another attack surface in addition to BMC and CPU surfaces. Similar system control issues as with BMC vulns.
  - Example: Intel ME/AMT password vuln (INTEL-SA-00075, CVE-2017-5689 (?))
- Unsure which of these stay running when CPU is powered off

# BMC Standards

- Intel IPMI
- DMTF SMASH
- DMTF DASH
- DMTF Redfish
- ...

# BMC Interfaces

- PECI (Platform Environment Control Interface)
- HECI (Embedded Controller Interface)
- PLDM (Platfform Level Data Model)
- PMCI (Platform Management Components Intercommunication)
- MCTP (Management Component Transport Protocol)
- NC-SI (Network Controller Sideband Interface)
- HP RIBCL (Remote Insight Board Command Language)
- SMbus
- ...

# BMC and NIC(s)

- How can IPMI (or Redfish or SMASH, etc.) do networking?
  1) Dedicated BMC NIC
  2) Piggyback on Host NIC
     - Cannot isolate BMC network traffic!
- Another related BMC net security issue:
  - Most use Ethernet-based NICs
  - But now a few vendors are starting to use WiFi-based NICs!
    - Cannot isolate BMC network traffic!
    - Some BMC protocols still expose username/password data in plaintext!
- Understand how a vendor implemented above before buying server

# BMCs and Mobile Apps

- At least one OEM makes a smartphone app that lets sysadmins remotely control systems using IPMI.
- Today, you have to secure access to BMC network traffic, probably via a sysadmin's desktop/laptop.
- Using a smartphone gives an attacker more ways to get control of your network. More attack surface to defend.
- How do you keep your BMC network traffic local if you send it through a phone carrier's network?
- IPMI protocols aside, similar issues with WWW protocols and Redfish, and WS-MAN-based management, they can be used on a mobile device. Similar issue with any BMC web interface.

# Vendor-Specific Implementations

- Each vendor has their own closed-source codebase:
  - HP: iLO
  - Dell iDRAC
  - Oracle/Sun: ILOM
  - AMI: MegaRAC
  - IBM/Lenovo: IMM, IMM2
  - Fujitsu iRMC
  - SuperMicro IPMI
  - …
- There is one main open source multi-vendor BMC project: OpenBMC

# Vendor-Specific Technologies

- Vendors extend an industry standard (IPMI, DASH, SMASH, Redfish) and add their own features
- Some of the extras network server/client protocols (incomplete list):
  - Telnet, SNMP, FTP, SMTP, VNC, SSH, LDAP, RADIUS, HTTP, HTTPS, SSH, WS-MAN, DHCP, RDP, AD, <vendor-centric network protocols>, ...

# OpenBMC

- OpenBMC is an open source Linux BMC implementation used by multiple vendors
- IBM uses it on OpenPOWER systems
- Yocto Linux-based
- IPMI support
- Redfish support in recent release
- Actively working on security issues!
- Related to the Open Compute Project

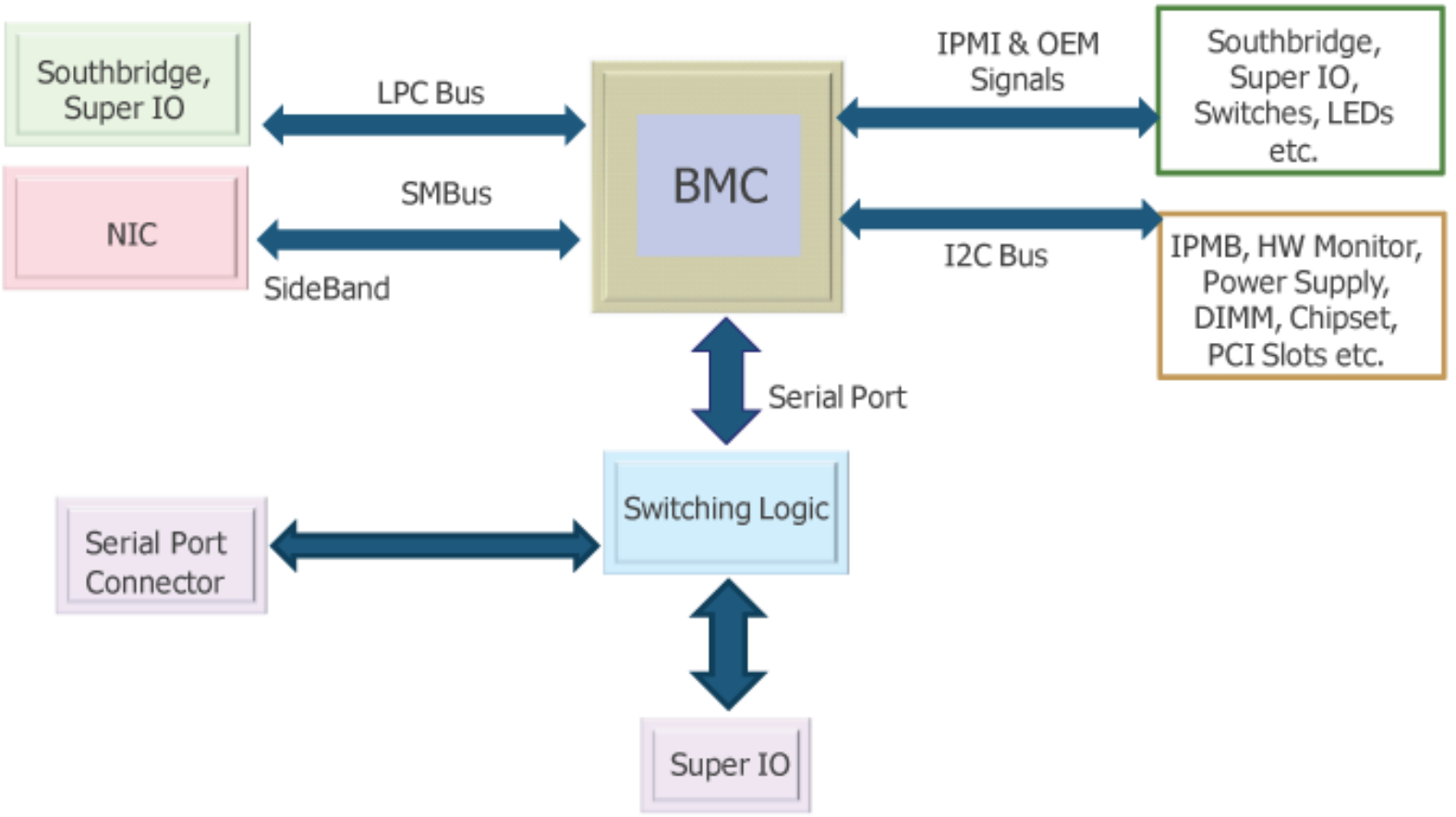
# Intelligent Platform Management Interface (IPMI)

- The IPMI specification defines a set of common interfaces to a computer system which system administrators use to monitor system health and manage the system.
- IPMI operates independently of the OS and allows administrators to manage a system remotely even without an OS, system management software, and even if the monitored system is powered off (along as it is connected to a power source). IPMI can also function after an OS has started, offering enhanced features when used with system management software
- Initial stakeholders: Dell, HP, Intel, and NEC
- V1.0 released on 1998-09-16, V1.5 on 2001-03-01, and V2.0 on 2004-02-14.
- Spec was “frozen” on March 2019

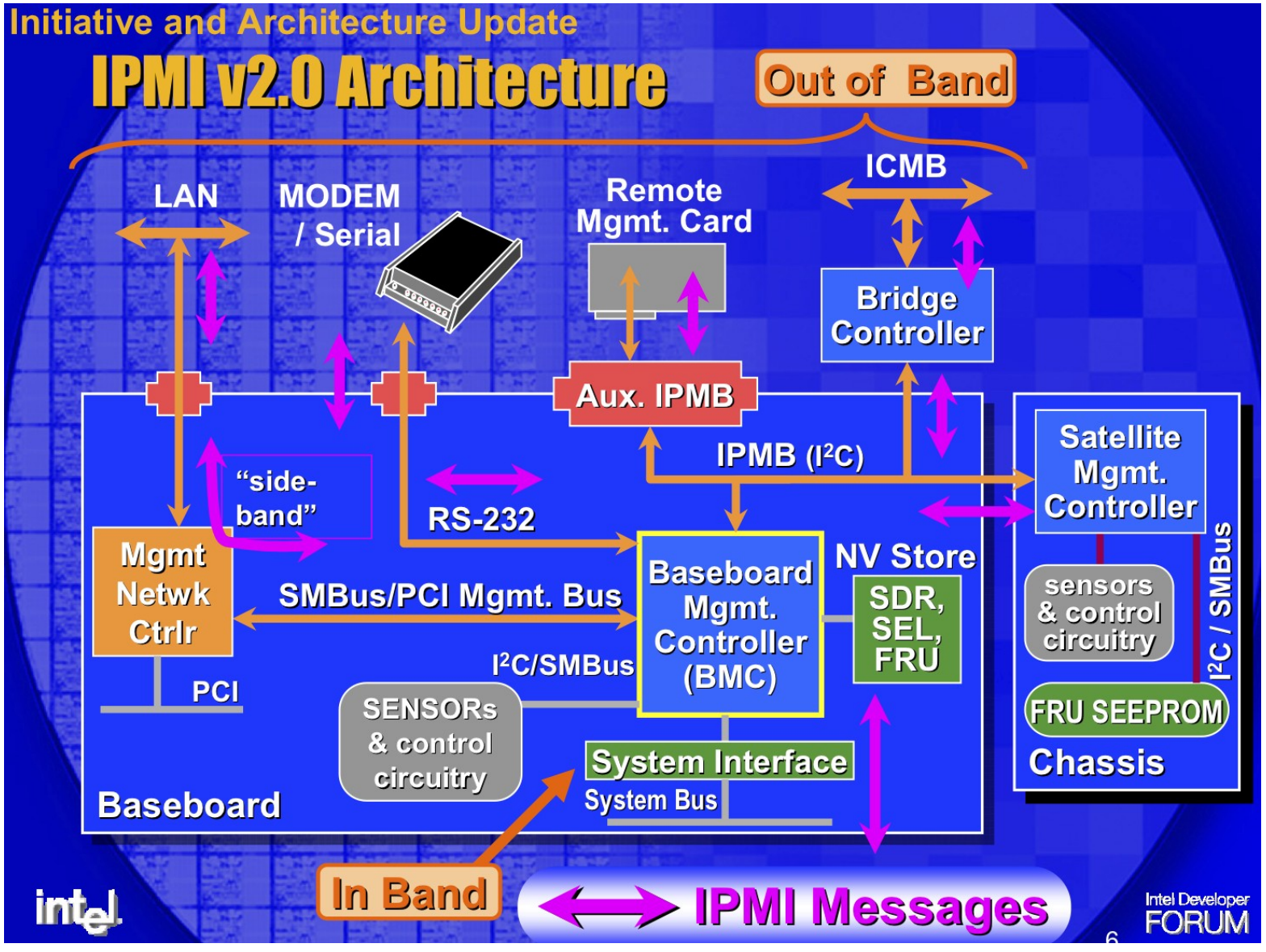
# IPMI Features

- IPMI Features:
  - Remote power control
  - Serial over LAN
  - Watchdog
  - Boot order
  - Sensor monitoring
  - Alarms

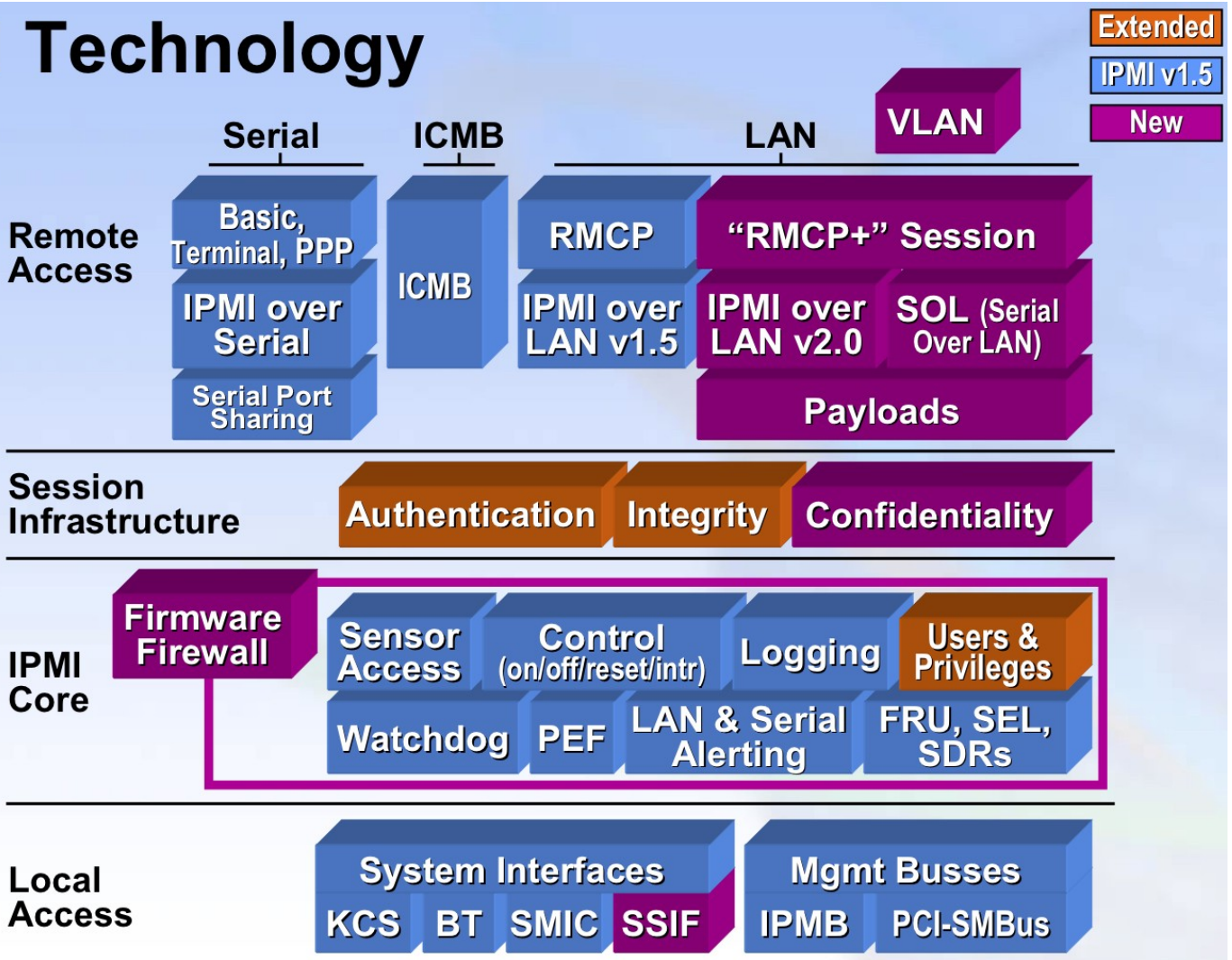
# IPMI Block Diagram



# IPMI v2.0 Architecture



# IPMI Technology




intel

# IPMI and SMM

- Modern Intel CPUs have a System Management Mode (SMM), in addition to their normal operating mode.
- IPMI talks about SMM, the usefulness of System Management Interrupts (SMIs) with IPMI. Attackers may be able to leverage SMM attacks with their IPMI attacks.
- CPU/SMM interaction is somewhat similar to CPU/BMC interaction:
  - SMM can see an do more with system than normal CPU mode,
  - like BMC can do more with system than the CPU.

# Some IPMI security issues

- IPMI 1.x:
  - NULL authentication option
  - NULL user option
  - Anonymous
  - UDP-based RMCP (Remote Management Control Protocol) vulnerable to multiple network security attacks (password sniffing, network spoofing, connection hijacking, MitM attacks, ...)
- IPMI 2.0:
  - Cipher Zero (passwords ignored!)
  - RAKP (RMCP+ Authenticated Key-Exchange Protocol) passwords
- Read Dan Farmer’s IPMI security research for details!!

# IPMI attack surface

- Once attacker has access to a BMC, they can compromise it's host server (firmware, OS)
- But they can also compromise other hosts
  - IPMI management groups share common passwords
- web interface: TLS certs, expired/revoked/self-signed/...
- Passwords stored on Flash or in RAM
  - IPMI passwords stored on Flash: Hardware Lifecycle Decommission/Disposition phases

# WS-Management

- Non-RESTful, WSDL/SOAP-based XML-based interfaces for managing systems, both for OS-present code as well as for OOB use.
- DMTF SMASH and DASH are WS-MAN-based
- Redfish uses RESTful-based interfaces

# Desktop and Mobile Architecture for System Hardware (DASH)

- DMTF's WS-Management-based system management standard for desktop and mobile client systems, providing out-of-band and remote management of desktop and mobile systems. DASH provides support for the redirection of KVM and text consoles, as well as USB and media, and supports the management of software updates, BIOS, batteries, NIC, MAC and IP addresses, as well as DNS and DHCP configuration

# Systems Management Architecture for Server Hardware (SMASH)

- DMTF's WS-Management-based standard for servers, provides the ability to remotely manage a platform independent of machine state, operating system state, server system topology or access method – interoperable management is possible before the OS is operational, when the OS is hung, or while the OS is up. SMASH specifies two interfaces, web services and command line. The web services are based on DMTF Web Services for Management (WS-Man). The command line is described in Server Management Command Line Protocol (SM CLP)

# Redfish

- DMTF’s Redfish is an extensible management standard using a data model representation inside of a hypermedia RESTful interface. Since it is model oriented, it is capable of expressing the relationships between components in modern systems as well as the semantics of the services and components within them. The model is exposed in terms of an interoperable Redfish Schema, expressed in an OData Schema representation with translations to a JSON Schema and OpenAPI representations, with the payload of the messages being expressed in a JSON following OData JSON conventions.

# Redfish Features

**Retrieve "IPMI class" data**

- Basic server identification and asset info
- Health state
- Temperature sensors and fans
- Power supply, power consumption and thresholds

**Basic I/O infrastructure data**

- Host NIC MAC address(es) for LOM devices
- Simple hard drive status / fault reporting

**Discovery**

- Service endpoint (network-based discovery)
- System topology (rack/chassis/server/node)

**Security**

- Session-based leverages HTTPS

**Perform Common Actions**

- Reboot / power cycle server
- Change boot order / device
- Set power thresholds

**Access and Notification**

- Serial console access via SSH
- Alert / event notification method(s)
- Event Log access method(s)

**BMC infrastructure**

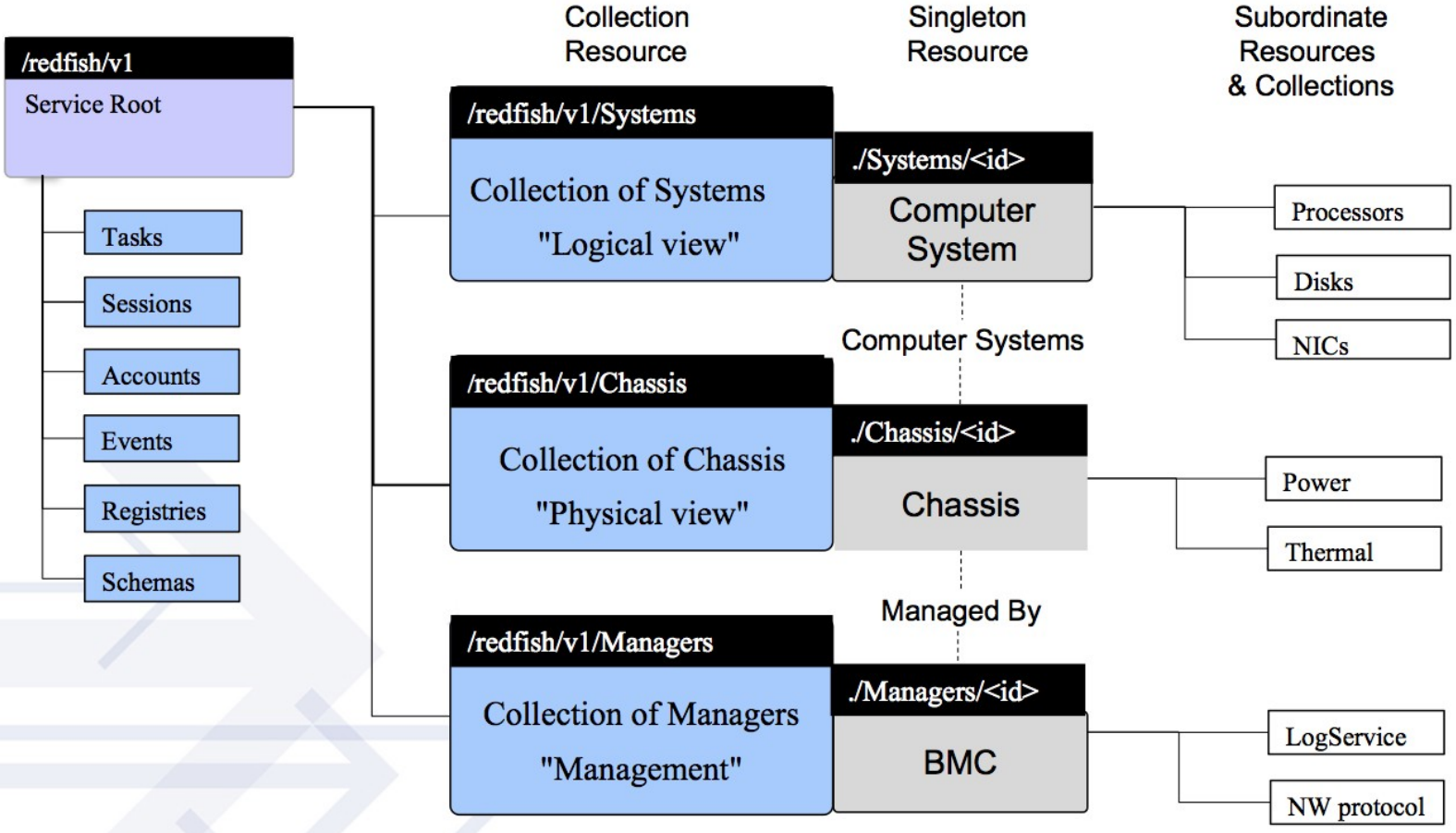
- View / configure BMC network settings
- Manage local BMC user accounts

**Working on more…**

# Redfish

- Redfish replaces (or at least competes with) SMASH, DASH, IPMI, and Intel AMT as the new LOM interface
- Redfish relies on HTTP/HTTPS for transport and OData-flavored JSON for data
- There are DMTF-based schemas, and vendor-centric ones
- AFAICT, most vendors are implementing Redfish using their existing LOM technologies (HP iLO, Dell DRAC, Lenovo IMM, etc.)
- OpenBMC just recently started supporting Redfish

# Redfish Resource Map (simplified)




**GET http://<ip-addr>/redfish/v1/Systems/{id}/Processors/{id}**

Use the Redfish Resource Explorer (redfish.dmtf.org) to explore the resource map

# DMTF Redfish tools on Github

| | Tool | Description |
|---|---|---|
| Extend | CSDL Validator | Validates the CSDL conforms to Redfish requirements |
| | CSDL-to-JSON schema convertor | Generates json-schema files from CSDL |
| | YANG to Redfish | Converts a YANG model into a set of Redfish CSDL files, enabling Ethernet switching standard access via Redfish |
| | Document Generator | Generates documentation from json-schema |
| Working Svc | Mockup Server | Exposes a mockup as a static HTTP service (GETs only) |
| | Mockup Creator | Creates a mockup from a Redfish service |
| | Profile Simulator | Dynamic simulator of the proposed Redfish profile for OCP |
| | Interface Emulator | Emulate a Redfish interface statically or dynamically. |
| Test | Service Validator | Validates a Redfish service is conformant |
| | JSON Schema Response Validator | Validates any JSON resource against DMTF provided JSON schemas |
| | Reference Checker | Validates the reference URLs in CSDL files |
| | Use Case Checker | Collection of tools to validate common use cases for Redfish Services. |
| | Service Conformance Tool | Verifies conformance of a Redfish service to assertions in the Redfish Specification |
| Client | CLI (redfishtool) | A command line tool for interacting with a Redfish service (similar to ipmitool) |
| | Event Listener | A lightweight HTTPS server that can be executed to read and record events from a Redfish Service |
| | C Library (libRedfish) | C libraries for interacting with Redfish services |
| | Python Utility & Library | A Command line tool with UI and python libraries for interacting with Redfish services |

# Redfish standards

- Redfish tries to leverage modern existing standards, including:
  - URI, HTTP, TLS
  - UPNP’s SSDP
  - HTTP-based alert subscription
  - JSON and OData v4, CSDL, JSON Schema

# Redfish: security

- There is very little existing documentation focusing on securing Redfish, it defers to existing HTTP/JSON/OData security best practices
- Redfish defers to WWW for security/authentication, see HTTPS guidance (OWASP, etc.) Look at the Redfish spec to see what HTTP headers/requests/responses are needed and any Redfish-centric assumptions.
- Treat Redfish as the most critical web app which your company needs to secure from attackers
- Isolate the network where Redfish traffic occurs
- Restrict access to network to only authenticated users.

# BMC Security Tools

- Metasploit IPMI modules
- IPMIPWN
- Ipmitool, OpenIPMI, FreeIPMI, ...
- IPMI password cracking: hashcat, john-the-ripper
- iLo_Toolkit (Redfish-based, HP iLO-centric)
- <vendor-centric-tools.....>
- <traditional embedded system/IoT tools...>

# BMC Hardening Checklists

- Understand which NIC(s) and what protocols your LOM solutions use
  - IPMI: read Dan Farmer’s Security Best Practices!!!
  - SMASH and DASH: use WWW/WS-Man/XML/SOAP/WSDL security best practices
  - Redfish: use existing WWW/HTTP/JSON/OData security best practices
- PreOS Security will have an upcoming BMC security best practices quick references shortly. https://preossec.com/

# Calls To Action

- Learn how to secure the BMCs of all systems you maintain
- DMTF Redfish WG and/or Web security community: help sysadmins Blue Teams with Redfish defensive best practices. Redfish tools are amongst the most important enterprise web apps that need securing, DMTF writes vendor-centric docs and need sysadmin-centric docs
- Sysadmin community: update documentation and certification training materials to add BMC security
- Intel: create an ME/AMT security best practices document

# More Information

- Vendor standards:
  - Intel IPMI
    - https://www.intel.com/content/www/us/en/servers/ipmi/ipmi-home.html
  - DMTF Redfish, https://redfish.dmtf.org/
  - DMTF DASH, https://dmtf.org/
  - DMTF SMASH, https://dmtf.org/
  - Intel Corp, “Intel® Sideband Technology: An Overview of the Intel Server Manageability Interfaces”
    - https://www.intel.com/content/www/us/en/embedded/products/networking/sideband-technology-appl-note.html

# More Information

- Sysadmin-focused documentation on using Redfish:
  - ADMIN Magazine: Redfish System Management
    - http://www.admin-magazine.com/Archive/2017/38/Redfish-standard-as-a-replacement-for-IPMI-Chassis-Management
  - Red Hat Summit 2018:
    - Using Ansible and Redfish to Automate Systems Management
    - https://www.youtube.com/watch?v=9tAXBvuyXJU
    - https://www.redhat.com/en/about/videos/summit-2018-using-ansible-and-redfish-automate-systems-management
  - OpenSource.com: Out-of-band management with Redfish and Ansible
    - https://opensource.com/article/17/9/out-band-management-redfish-and-ansible

# More Information

- Industry security guidance:
  - US-CERT: Risks of Using the Intelligent Platform Management Interface (TA13-207A), https://www.us-cert.gov/ncas/alerts/TA13-207A
  - Cisco: IPMI Security Vulnerabilities, https://www.cisco.com/c/en/us/about/security-center/ipmi-vulnerabilities.html
  - IBM: IPMI best practices, https://www.ibm.com/support/knowledgecenter/en/P9ESS/p9eih/p9eih_ipmi_bestpractices.htm
  - SuperMicro, “Best Practices: BMC Security”, https://www.supermicro.com/products/nfo/files/IPMI/Best_Practices_BMC_Security.pdf

# More Information

- Security research:
  - Dan Farmer, IPMI research, http://fish2.com/ipmi/
  - Dan Farmer, “IPMI++ Security Best Practices”, http://fish2.com/ipmi/bp.pdf
  - HD Moore, “A Penetration Tester's Guide to IPMI and BMCs”, https://blog.rapid7.com/2013/07/02/a-penetration-testers-guide-to-ipmi/
  - Joffrey Czarny, Alexandre Gazet, Fabien Perigaud, "Subverting your server through its BMC: the HPE iLO4 case", https://recon.cx/2018/brussels/talks/subvert_server_bmc.html
  - Matias Soler, Nico Waisman, 'The Unbearable Lightness of BMC’s", https://www.blackhat.com/us-18/briefings/schedule/index.html#the-unbearable-lightness-of-bmcs-10035
  - Jesse Michael, Mickey Shkatov, Oleksandr Bazhaniuk, "Remotely Attacking System Firmware", https://www.blackhat.com/us-18/briefings/schedule/index.html#remotely-attacking-system-firmware-11588
  - Joffrey Czarny, Alexandre Gazet, Fabien Perigaud, "Backdooring your server through its BMC: the HPE iLO4 case", https://www.sstic.org/2018/presentation/backdooring_your_server_through_its_bmc_the_hpe_ilo4_case/
  - Joffrey Czarny, Alexandre Gazet, Adrien Guinet, Fabien Perigaud, “Defeating NotPetya from your iLO4”, https://airbus-seclab.github.io/ilo/Whitepaper-Defeating_NotPetya_from_your_iLO4-guinet-perigaud-gazet-czarny.pdf

# Questions?

- Comments? Questions?
- Thanks for attending!

- Slides will be posted in an upcoming blog post to FirmwareSecurity.com in a day or two, thanks for your patience.

# Image Credits

- All embedded graphics were done by others, I snipped them for this presentation. I’m grateful for their graphic abilities.
- IPMI graphics:
  - Wikipedia, IPMI
  - Advances in Intelligent Platform Management: Introducing the New IPMI v2.0 Specifications, Tom Slaight, Intel, 2004/02, Intel IDC
- Redfish graphics:
  - <FIXME: info on dmtf.org-based slides on Redfish>